REMARKS

In the Office Action, the Examiner noted that Claims 1 through 27 were pending in the Application. The Examiner rejected Claims 1-16, 24, 26 and 27 and allowed Claims 17-23 and 25. Claim 26 has been canceled in this Amendment. Thus, Claims 1 - 25 and 27 are now pending in the Application. Applicant traverses the rejections below.

I. Traversal of the Rejection under 35 USC Section 101

The Examiner rejected claims 1-12 and 26-27 under 35 USC 101.

Independent Claims 1, 5, and 27 are method claims; Claim 26 has been canceled. Claims 1, 5 and 27 have been amended to affirmatively indicate that they are computer implemented. Amending the claims in this manner should place them in conformance with the requirements of 35 USC section 101, along with their dependent claims. Withdrawal of the rejection is respectfully requested.

II. Traversal of the Rejection over the Cited Art

The Examiner rejected Claims 1, 4, 12, 13, 16, 24 and 26 under 35 U.S.C. 102(a) as being anticipated by the article by Passerone et al (Passerone). The Examiner rejected Claims 2, 3, 14 and 15 as being unpatentable over Passerone in view of the article by Lobe et al (Lobe). Applicant traverses this rejection below.

A. The Present Invention

The present invention discloses a technique for generating mapping source code to establish mapping connections between enterprise system nested array object fields and legacy

system nested array object fields is disclosed. For each desired mapping connection in a received list of desired connections, a determination of an enterprise system field and legacy system field to be mapped, as well as a connection nesting level, is made. The identity of the system arrays containing the enterprise and legacy system fields is also determined. A logical tree is created which includes a root node, one leaf node for each desired connection, and, for each leaf node, N intermediate nodes interconnecting the leaf node with the root node, where N is equivalent to the determined nesting level of the connection associated with that leaf node, and where each of the N intermediate nodes that is successively further from the root node is associated with a successively more deeply nested target system array. The tree is then traversed depth-first to generate the desired mapping source code.

B. Differences Between the Present Claims and the Cited Art

Passerone describes a procedure for minimizing the code size of sequential programs for reactive systems. It identifies repeated code segments and finds a minimal covering of the input control flow graphs with code segments. The segments do not have the same code in common. This is quite different from the present claimed invention. Passerone relies on the use of a schedule graph, which is "a finite directed graph, where each node has a unique name and each arc corresponds to an action."

Claim 1 recites a "method of generating program source code in a computer environment to perform a mapping task in which enterprise system nested array object fields and legacy system nested array object fields are mapped to one another". The cited passages from Passerone do not teach, suggest or disclose the concept of mapping, much less mapping enterprise system nested array object fields and legacy system nested array object fields to one another. There is no discussion of legacy or enterprise issues in Passerone. Rather, Passerone minimizes the code size of sequential programs for reactive systems. Claim 1 further discusses "performing a depth-first traversal of a logical tree having a root node, a leaf node for each desired mapping connection,

and intermediate nodes....each intermediate node being associated with an array". There is no discussion or teaching in Passerone of a leaf node for each desired mapping connection or intermediate nodes associated with an array.

Claim 1 also recites "for each intermediate node visited when traversing away from said root node, generating program source code to open a loop; for each visited leaf node, generating program source code to create the mapping connection represented by said visited leaf node; and for each intermediate node having no unvisited children that is visited when traversing towards said root node, generating program source code to close said loop." Relative to this subject matter, Passerone does not teach, suggest or disclose creating a mapping connection, closing a loop, or opening a loop. Once again, Passerone is directed to minimizing the code size of sequential programs, not performing a mapping task, as per the present claims.

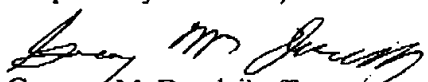
Accordingly, Applicant submits that independent Claim 1 patentably distinguishes over Passerone. Independent Claims 13 and 24 were rejected for the same reasons as Claim 1, so it follows that these claims, as well as dependent Claims 4, 12, and 16, also patentably distinguish over Passerone.

Dependent Claims 2, 3, 14 and 15 were rejected over rejected over the combination of Passerone and Lobe. Since it has been shown above that their independent claims patentably distinguish over Passerone, it follows that these claims patentably distinguish over the cited combination.

III Summary

Applicant has presented technical explanations and arguments fully supporting their position that the pending claims contain subject matter which is not taught, suggested or disclosed by Passerone, Lobe, or any combination thereof, and has amended the claims to place them in conformance with the requirements of Section 101. Accordingly, Applicant submits that the present Application is in a condition for Allowance. Reconsideration of the claims and a Notice of Allowance are earnestly solicited.

Respectfully submitted,



Gregory M. Doudnikoff
Attorney for Applicant
Reg. No. 32,847

GMD:dl

Docket No: CA9-00-078
PHONE: 919-254-1288
FAX: 919-254-4330